

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

(19)



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: 11355150 A

(43) Date of publication of application: 24 . 12 . 99

(51) Int. Cl.

H03M 13/12  
G06F 11/10

(21) Application number: 10160345

(22) Date of filing: 09 . 06 . 98

(71) Applicant: SONY CORP

(72) Inventor: HATAKEYAMA IZUMI

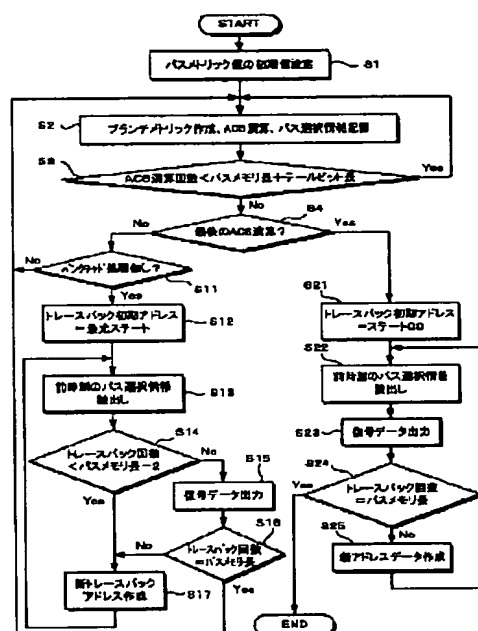
## (54) PUNCTURED VITERBI DECODING METHOD

(57) Abstract:

**PROBLEM TO BE SOLVED:** To improve decoding precision without lengthening a pass memory for a trace-back.

**SOLUTION:** ACS calculation based on reception data is executed in steps S1-S4. Besides, a step concerning the trace-back after a step S12 is executed by a step S11 only at the time of no punctured processing. Since a punctured pattern where data for the portion of one time point among reception data for the portion of three time points is not punctured is used here, the trace-back corresponding to data for the portion of one time point is executed. Then, decoding data for the portion of three time points including punctured data for the portion of two time points is generated by one time trace-back. Decoding error occurrence is prevented or reduced by not executing the trace-back with punctured data as a start point, which has a high probability of causing an erroneous decoding processing result.

COPYRIGHT: (C)1999,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-355150

(43) 公開日 平成11年(1999)12月24日

(51) Int.Cl.<sup>6</sup>

H 0 3 M 13/12

G 0 6 F 11/10

識別記号

3 3 0

F I

H 0 3 M 13/12

G 0 6 F 11/10

3 3 0 N

審査請求 未請求 請求項の数 7 O L (全 14 頁)

(21) 出願番号

特願平10-160345

(22) 出願日

平成10年(1998)6月9日

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 畠山 泉

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

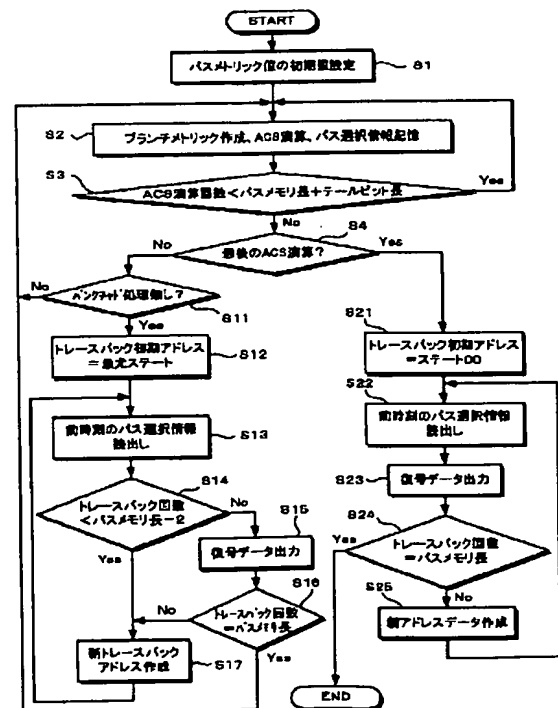
(74) 代理人 弁理士 杉浦 正知

(54) 【発明の名称】 パンクチャドビタビ復号方法

(57) 【要約】

【課題】 パンクチャドビタビ復号方法において、トレースバックを行うバスメモリの長さを長くせずに復号精度を向上させる。

【解決手段】 ステップS1～ステップS4により、受信データに基づくACS演算を行う。さらに、ステップS11により、パンクチャド処理が無い場合にのみステップS12以降のトレースバックに係るステップを行うようになされる。ここでは3時点分の受信データの内、1時点分のデータがパンクチャドされないようなパンクチャド・パターンが使用されるので、その1時点分のデータに対応するトレースバックが行われる。そして、1回のトレースバックにより、パンクチャドされた2時点分のデータをも含む3時点分の復号データが生成される。誤った復号処理結果を招く可能性が高い、パンクチャドされたデータを起点とするトレースバックを行わないことにより、復号誤りの発生を防止または低減することができる。



## 1

## 【特許請求の範囲】

【請求項 1】 畳込み符号化したデータに、所定の周期で一部を間引く処理を施すことによって生成されたデータに基づいて、ビタビアルゴリズムを用いた最尤復号を行うパンクチャドビタビ復号方法において、

受信したデータについて、所定数のシンボルからなる復号処理単位毎にデータが間引かれているか否かを検出するパンクチャド制御ステップと、

上記受信したデータに基づいて、上記復号処理単位毎にブランチメトリックの値を計算するブランチメトリック作成ステップと、

上記ブランチメトリック作成ステップの結果に基づいて、畳込み符号化に関連して予め決められた複数のステートに対してパスメトリックの値を計算し、計算した上記パスメトリックの値に基づいて最尤な状態遷移を選択して選択結果を示すパス選択情報を生成する A C S 演算を行う A C S 演算ステップと、

上記パスメトリックの値と、上記パス選択情報とに基づいて復号データを生成するためのトレースバックを行うトレースバック処理ステップと、

上記パンクチャド制御ステップによってデータが間引かれていると判定された上記復号処理単位に基づいて初期値を設定して上記トレースバック処理ステップを実行することをせず、上記パンクチャド制御ステップによってデータが間引かれていないと判定された上記復号処理単位に基づいて初期値を設定して上記トレースバック処理ステップを実行するように制御するトレースバック可否制御ステップとを有することを特徴とするパンクチャドビタビ復号方法。

【請求項 2】 請求項 1 において、

上記トレースバック可否制御ステップによって上記トレースバック処理ステップを実行する旨の制御がなされる際に、上記パンクチャド制御ステップによってデータが間引かれていないと判定された上記復号処理単位に対応する復号データに加えて、上記パンクチャド制御ステップによってデータが間引かれていると判定された上記復号処理単位に対応する復号データを生成するようにしたことを特徴とするパンクチャドビタビ復号方法。

【請求項 3】 請求項 1 において、

上記トレースバック可否制御ステップは、上記パンクチャド制御ステップによってデータが間引かれていると判定された上記復号処理単位が供給されるタイミングにおいては上記 A C S 演算ステップを実行せず、上記パンクチャド制御ステップによってデータが間引かれていないと判定された上記復号処理単位が供給されるタイミングにおいて上記 A C S 演算ステップを実行するように制御する A C S 演算可否制御ステップをさらに有することを特徴とするパンクチャドビタビ復号方法。

【請求項 4】 請求項 3 において、

## 2

上記トレースバック可否制御ステップによって上記トレースバック処理ステップおよび上記 A C S 演算ステップを実行する旨の制御がなされる際に、上記パンクチャド制御ステップによってデータが間引かれていないと判定された上記復号処理単位に対応して上記 A C S 演算および復号データ出力を行うことに加えて、上記パンクチャド制御ステップによってデータが間引かれていると判定された上記復号処理単位に対応して上記 A C S 演算および復号データ出力を行うようにしたことを特徴とするパンクチャドビタビ復号方法。

【請求項 5】 請求項 1 に記載されたパンクチャドビタビ復号方法であって、

所定量のデータ毎に処理が完結し、処理が完結する毎に畳込み符号器を初期化するようにした畳込み符号化によって生成されるデータを復号するパンクチャドビタビ復号方法において、

畳込み符号化における畳込み符号系列中で、畳込み符号器の出力が固定値 '0' となる場合に計算される各ステートのパスメトリック値を参照して重み付けがなされた値を、上記 A C S 演算ステップにおいて各初期ステートのパスメトリック値として設定することを特徴とするパンクチャドビタビ復号方法。

【請求項 6】 請求項 1 に記載されたパンクチャドビタビ復号方法であって、

所定量のデータ毎に処理が完結し、処理が完結する毎に符号化したデータの最後にテールビットを付加するようにした畳込み符号化によって生成されるデータを復号するパンクチャドビタビ復号方法において、

上記受信したデータの量が上記トレースバック処理ステップの処理単位である所定のデータ量よりも少ない場合に、全てのデータに基づく上記 A C S 演算ステップを行った後に、テールビットに対応するステートを起点として上記トレースバック処理ステップを行うことを特徴とするパンクチャドビタビ復号方法。

【請求項 7】 請求項 1 に記載されたパンクチャドビタビ復号方法であって、

所定量のデータ毎に処理が完結し、処理が完結する毎に符号化したデータの最後にテールビットを付加するようにした畳込み符号化によって生成されるデータを復号するパンクチャドビタビ復号方法において、

上記受信したデータの量が上記トレースバック処理ステップの処理単位である所定のデータ量よりも多い場合に、上記 A C S 演算ステップと、上記トレースバック処理ステップとを行い、復号すべき残りのデータの量が上記トレースバック処理ステップの処理単位である所定のデータ量より小さくなった時に、上記残りのデータに基づく上記 A C S 演算ステップを行った後に、テールビットに対応するステートを起点として上記トレースバック処理ステップを行うことを特徴とするパンクチャドビタビ復号方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明は、例えばデジタル移動体通信機に使用されるパンクチャドビタビ復号方法に関する。

## 【0002】

【従来の技術】ビタビ復号等の復号精度の高い復号方法を使用する情報伝送においては、送信側で畳込み符号を所定の比率で間引くパンクチャド処理を行うことによって伝送されるデータの単位量当たりの情報量を増加させることが行われる。例えば、畳込み符号6ビット毎に2ビットを間引くようなパンクチャド処理を行う場合には、伝送されるデータの単位量当たりの情報量を1.5倍に増加させることができる。

【0003】従来、パンクチャド処理された畳込み符号化データをビタビアルゴリズムを使用して最尤復号する復号方法を行う場合には、パンクチャドしたデータ部分を含むパンクチャド処理を行うブランチメトリック作成時、およびパンクチャドしたデータ部分を含まないパンクチャド処理無しのブランチメトリック作成時の何れにおいても、以下のような復号処理が行われる。

【0004】すなわち、ACS演算を含む復号処理を開始後、所定の処理時点において到達した最尤スタートを起点としてトレースバックを行い、所定長分戻った時点での復号処理データに基づいて復号データを生成する。このような処理を各処理時点において順次行い、必要とされる復号データ分処理を行って復号していた。

## 【0005】

【発明が解決しようとする課題】このような復号処理において、パンクチャドした畳込み符号化データ自体に誤りが多く発生している場合には、パンクチャドされているデータに対応するブランチメトリックを作成するための演算処理の対象とされるデータが誤ってしまう可能性が高くなる。その結果として、パンクチャドされているデータに対応するブランチメトリックの値、さらにはブランチメトリックの値に基づいて行われるACS演算の結果として得られる最尤スタートが大きく誤ってしまう確率が高くなる。

【0006】そこで、トレースバックによる正しいパスへの収束を担保し、復号精度を向上させるために、トレースバックを行うパスメモリ長さを長くする必要が生じる。但し、そのような構成を用いると、回路規模が増大する、高速なクロックが必要とされる、あるいは処理の高速化が妨げられる等の問題が生じていた。

【0007】従って、この発明の目的は、トレースバックを行うパスメモリの長さを長くしなくても、復号精度を向上させることが可能なパンクチャドビタビ復号方法を提供することにある。

## 【0008】

【課題を解決するための手段】請求項1の発明は、畳込

み符号化したデータに、所定の周期で一部を間引く処理を施すことによって生成されたデータに基づいて、ビタビアルゴリズムを用いた最尤復号を行うパンクチャドビタビ復号方法において、受信したデータについて、所定数のシンボルからなる復号処理単位毎にデータが間引かれているか否かを検出するパンクチャド制御ステップと、受信したデータに基づいて、復号処理単位毎にブランチメトリックの値を計算するブランチメトリック作成ステップと、ブランチメトリック作成ステップの結果に基づいて、畳込み符号化に関連して予め決められた複数のステートに対してパスメトリックの値を計算し、計算したパスメトリックの値に基づいて最尤な状態遷移を選択して選択結果を示すパス選択情報を生成するACS演算を行うACS演算ステップと、パスメトリックの値と、パス選択情報とに基づいて復号データを生成するためのトレースバックを行うトレースバック処理ステップと、パンクチャド制御ステップによってデータが間引かれていると判定された復号処理単位に基づいて初期値を設定してトレースバック処理ステップを実行することをせず、パンクチャド制御ステップによってデータが間引かれていないと判定された復号処理単位に基づいて初期値を設定してトレースバック処理ステップを実行するように制御するトレースバック可否制御ステップとを有することを特徴とするパンクチャドビタビ復号方法である。

【0009】以上のような発明によれば、パンクチャドされたデータに基づいて設定される初期値の下でのトレースバックが行われなくすることができる。

【0010】このため、パンクチャドされたデータに基づいて設定される初期値の下でのトレースバックが行われることにより、畳込み符号中の誤りに起因して最尤パスとして誤ったものが選られ、その結果として復号誤りが生じることを防止または低減することができる。

## 【0011】

【発明の実施の形態】以下、デジタル移動体通信機の端末に対してこの発明を適用したこの発明の一実施形態について説明する。まず、この発明の一実施形態の全体構成について図1を参照して説明する。アンテナ100を介してベース局との無線データの入出力が行われる。アンテナ100には送受信共用器101が接続されている。送受信共用器101は、受信した無線データを受信機102に供給し、また、送信すべき無線データを送信機104から受取る。

【0012】受信側の構成についてまず説明する。受信機102は、供給される無線データを復調し、チャンネルデコーダ105に供給する。チャンネルデコーダ105は、供給される信号に音声および通信制御情報のフォーマット変換および復号化処理を施し、処理結果を音声コーデック107に供給する。音声コーデック107は、チャンネルデコーダ105の出力を音声情報信号に変換す

10

20

30

40

50

## 5

る復号化処理を行い、音声情報信号を送受話器108に供給する。送受話器108は、供給される情報に基づいてスピーカ111を駆動し、音声を発生させる。

【0013】次に、送信側の構成について説明する。マイク112は、入力する音声を所定の信号に変換して送受話器108に供給する。送受話器108は、供給される信号に基づいて音声情報信号を生成し、この音声情報信号を音声コーデック107に供給する。音声コーデック107は、音声情報信号に符号化処理を施し、処理結果をチャンネルエンコーダ106に供給する。チャンネルエンコーダ106は、供給される信号に音声および通信制御情報のフォーマット変換および符号化処理を施し、処理結果を送信機104に供給する。送信機104は、供給される信号を無線デックとしてキャリア上に重畳する処理を行い、処理結果を送受共用器101に供給する。

【0014】また、シンセサイザ103は送受信周波数を制御する。一方、コントローラ113は、通信制御情報の作成／解読と共にその設定／解除／維持を行う。さらに、コントローラ113は、シンセサイザ103の制御、およびキー／ディスプレイ部109の1／0制御等をも行う。ここで、キー／ディスプレイ部109は、例えばユーザが送信先を設定するためのキー入力部、設定した送信先等の情報を表示する例えば液晶表示パネル等の表示部等を有する。なお、以下の説明においては、チャンネルデコーダ105およびチャンネルエンコーダ106をチャンネルコーデック200と総称する。

【0015】図2を参照して、チャンネルコーデック200について詳細に説明する。まず、チャンネルエンコーダ106は、CRCジェネレータ204、畳込み符号器205、インターリーブ206を有する。これらの構成要素は、コントローラ113の指令に従って動作する。音声コーデック107から供給される音声情報信号およびコントローラ113から供給される制御情報信号がCRC(Cyclic Redundancy Code)ジェネレータ204に供給される。

【0016】CRCジェネレータ204は、供給される信号に後述するようにしてCRCを付加し、CRCを付加した信号を畳込み符号器205に供給する。畳込み符号器205は、供給される信号に誤り訂正用の符号化処理を施して畳込み符号を生成し、この畳込み符号をインターリーブ206に供給する。インターリーブ206は、供給される畳込み符号に、インターリーブ処理を施して送信機104に供給する。

【0017】インターリーブ処理および後述するデイン

$$g(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^4 + x + 1 \quad (1)$$

このCRCビット12ビットが情報ビットに付加されて280ビットを単位とする信号に変換される。さらに、値が'0'のテールビット8ビットが付加されて288ビットを単位とする信号とされる。ここで、テールビットは、畳込み符号中の最後の(拘束長-1)個のビットを

## 6

ターリーブ処理を行うことにより、伝送されるデータ中にバースト誤りが生じる場合にも、復号処理の対象とされるデータにおいてランダムな誤りのみが含まれるようにすることができる。これらの処理により、特に、ビタビ復号において復号エラーレートの大幅な低下の原因となり得るバースト誤りを除去することができる。

【0018】一方、チャンネルデコーダ105は、デインターリーブ201、ビタビ復号器202、CRCチェッカ203を有する。これらの構成要素は、コントローラ113の指令に従って動作する。受信機102から出力される受信デックがデインターリーブ201に供給される。デインターリーブ201は、供給される信号にデインターリーブ処理を施して、デインターリーブ処理によって生成される信号をビタビ復号器202に供給する。

【0019】ビタビ復号器202は、供給される信号に誤り訂正処理を施し、かかる処理の結果をCRCチェッカ203に供給する。CRCチェッカ203は、供給される信号について、データとしての正誤をCRCを参照して検出する。CRCチェッカ203の出力の内、音声情報に係る信号が音声コーデック107に供給され、一方、制御データに係る信号がコントローラ113に供給される。

【0020】次に、この発明の一実施形態において使用されるデータフォーマットについて図3を参照して説明する。図3に示すデータフォーマットの一例は、1周期20msecのフレームを単位として無線回線を通して通信する。伝送速度は9600bps、インターリーブ後のシンボル数は384シンボルとされる。また、ここでは、6シンボルがバンクチャドされて4シンボルとされる。そして、畳込み後のシンボル数は576シンボルとされる。さらに、畳込み符号化率は1/2、20msec当たりのデータ量は288ビットとされる。この288ビットの内、音声情報等に係る情報ビットが268ビットを占め、後述するようなCRC、テールビットがそれぞれ、12ビット、8ビットである。

【0021】次に、送信系および受信系での処理について図4を参照して詳細に説明する。まず、送信系の処理について説明する。上述したようにして生成される288ビットを単位とする情報ビットがCRCジェネレータ204に供給される。CRCジェネレータ204は、例えば以下の式(1)のような生成多項式に従って12ビットからなるCRCビットを生成する。

【0022】

畳込み符号器から出力させるために畳込み符号の末尾に付加されるものであり、(拘束長-1)のビット長を有するものとされる。そして、テールビットが付加された信号が拘束長 $k=9$ 、符号化率 $R=1/2$ の畳込み符号器205に供給される。畳込み符号器205は、供給さ

## 7

れる信号に後述するような畳込み符号化処理を施して、576ビットを単位とする信号に変換する。

【0023】畳込み符号器205が生成する畳込み符号がパンクチャド処理部207に供給される。パンクチャド処理部207は、畳込み符号に対してパンクチャド処理、すなわち、例えば以下の(2)のようなパターンに従うデータの間引き処理を施し、2/3の符号化を行った場合に相当する384ビットのデータに変換する。

【0024】

パンクチャリング・パターン：“110101”

“0”：削除、“1”：使用

(2)

(2)のパターンに従うパンクチャド処理は、576ビットデータを6ビットずつに分割し、その6ビットの内、1番目、2番目、4番目、6番目のデータを使用し、3番目、5番目のデータを間引く、すなわち使用しないで除くことにより、6ビットのデータを4ビットのデータに変換するものである。これにより、576ビットのデータが384ビットのデータに変換される。この384ビットのデータがインターリーブ206に供給され、データの順番が入替えられた後に送信機104に供給される。送信機104は、供給されるデータを無線回線に出力する。

【0025】一方、受信系での処理について詳細に説明する。受信機102が無線データを受信し、1フレーム当たり384シンボルのデータをデインターリーブ201に供給する。デインターリーブ201は記憶回路を備えており、この記憶回路に供給されるデータを所定量保持した後にデインターリーブを行ってデータの順番を入替える。これにより、送信系でインターリーブする前のデータの順番に戻して出力する。デインターリーブ201の出力は、ビタビ復号器202に供給される。

【0026】なお、この発明の一実施形態では、フレーム内で完結する同一フレーム内のビットインターリーブを採用している。そこで、デインターリーブ201では、フレームを単位とした処理を行うことが可能となるまで記憶回路にシンボルデータを保持した後、デインターリーブを行う。

【0027】ビタビ復号器202は、デインターリーブ201の出力に基づいて後述する復号処理を行い、復号データを生成する。この復号処理において、パンクチャド制御部210から供給される、パンクチャドされているデータタイミングを参照して後述するような処理が行われることにより、パンクチャドされているデータ位置においても、尤度の高い復号処理が可能となる。復号データがCRCチェック部203に供給され、上述した式(1)等の多項式に従って生成・付加された12ビットのCRCに基づくCRCチェックが行われる。

【0028】次に、図5を参照して畳込み符号器205について説明する。畳込み符号器205は、1クロック分の遅延を生じる8個のレジスタD1、D2、D3、D

## 8

4、D5、D6、D7、D8を有し、また、例えばmod2の乗算器等の演算器g0、g1を有する。そして、供給される信号中の各ビットと共に、レジスタD1、D2、D3、D5、D7、D8の出力が演算器g0に供給される。また、供給される信号中の各ビットと共に、レジスタD2、D3、D8の出力が演算器g1に供給される。演算器g0が供給される各データを演算して符号化シンボル出力C0を出力し、一方、演算器g1が供給される各データを演算して符号化シンボル出力C1を出力する。これらの符号化シンボル出力によって畳込み符号が構成される。

【0029】次に、ビタビ復号器202による復号処理について説明する。この発明の一実施形態では畳込み符号/ビタビ復号方式を使用しているため、復号処理の対象とされるデータが軟判定データとして処理される。この発明の一実施形態では、8値軟判定データとして処理される。一般に、ビタビ復号器は、入力するデータに基づいてビタビアルゴリズムに沿う計算処理を行うことによって、各ステート(状態)に至る例えば2つのパスの内の最尤なものを生き残りパスとして選択し、生き残りパスの選択が所定のパス長まで完了する毎に、生き残ったパスの系列の中で最尤なものを検出することにより、受信信号を復号するものである。ここで、各ステートは、符号化の方法に応じて予め規定される。例えば、この発明の一実施形態では拘束長k=9、符号化率R=1/2の畳込み符号が使用されるが、このような拘束長k=9の畳込み符号に対する復号処理においては、 $2^{k-1} = 2^{9-1} = 256$ 個のステートが用いられる。また、この発明の一実施形態では、パス選択情報の記憶段数(すなわち過去のパス選択情報の記憶時刻数)を64段としている。

【0030】ビタビ復号器202においては、384シンボルを単位とするデータが576シンボルを単位とするデータに変換され、さらに、かかるデータに基づいて、ビタビアルゴリズムを使用して拘束長k=9、符号化率R=1/2の最尤復号を行い、280ビットのデータが復号される。

【0031】図6を参照してビタビ復号器202の構成および動作についてより具体的に説明する。ビタビ復号器202は、ブランチメトリック演算部300、ACS(Add Compare Select)演算部302、パスメトリック記憶部301、パス選択情報記憶部304、最尤検出部303、およびデータ推定部305を有する。パス選択情報記憶部304内には、記憶段数が例えば64段の記憶回路が備えられている。

【0032】ブランチメトリック演算部300は、クロックに従う各時点での受信データと各ブランチとの間のブランチメトリック(例えばユークリッド距離等)の値を計算し、計算したブランチメトリックの値をACS演算部302に供給する。ACS演算部302は、供給さ

れるブランチメトリックの値に基づいて後述するACS演算を行って各ステートに対するパスメトリックの値を計算する。そして、計算したパスメトリックの値に基づいて生き残りパスを選択すると共に、計算したパスメトリックの値をパスメトリック記憶部301および最尤検出部303に供給する。また、生き残りパスの選択結果を示すパス選択情報を生成して、パス選択情報をパス選択情報記憶部304に供給する。

【0033】パスメトリック記憶部301は、供給される各ステートでのパスメトリックの値を記憶する。後述するように、記憶されたパスメトリックの値が1クロック後のACS演算を行うために使用される。また、パス選択情報記憶部304は、パス選択情報（パスの推定出力）を記憶する。最尤検出部303は、各ステートに対するパスメトリックの値に基づいて最尤ステート情報を生成して、この最尤ステート情報をデータ推定部305に供給する。データ推定部305は、最尤ステート情報と、パス選択情報記憶部304から読出した過去のパス選択情報とを使用してデータ推定を行い、復号データを出力する。

【0034】次に、ブランチメトリック演算部300によるブランチメトリックの生成についてより詳細に説明する。図3を参照して上述したような伝送速度9600bpsのフォーマットの場合、ブランチメトリック演算部300は、供給されるデータ（384シンボルを単位とする）に基づいて、2シンボルを単位とする処理を行うことにより、ブランチメトリックを作成する。この際に、パンクチャ制御部210が作成する、パンクチャドされたデータのタイミングを参照して、パンクチャドされたデータのための補正を行う。

【0035】ブランチメトリック演算部300は、連続する2シンボルの極性が'00','01','10','11'である場合に対応してBM(0,0), BM(0,1), BM

$$BM(0,0) = BM0(A) + BM0(B) \quad (3)$$

$$BM(0,1) = BM0(A) + BM1(B) \quad (4)$$

$$BM(1,0) = BM1(A) + BM0(B) \quad (5)$$

$$BM(1,1) = BM1(A) + BM1(B) \quad (6)$$

ここで、パンクチャ制御部210からパンクチャドされたデータタイミングが供給される時には、該当するシンボルデータに対応するBM0, BM1を共に"0"とする。このようにすれば、パンクチャドされたデータ部分に対応して各ステートに対するブランチメトリックの値が相対的に変わらないことになる。かかる状況は、パンクチャドされたデータ部分には情報が無いことに対応するものである。なお、BM0, BM1を共に"1"としても良い。

【0040】従って、式(3)～(6)において、例えばデータAがパンクチャドされたデータタイミングに対応するものである場合には、BM0(A), BM1

(A)が何れも"0"とされるので、ブランチメトリック

(1,0), BM(1,1)を作成する。ここで、2シンボルを単位として処理を行うのは、上述したフォーマットにおいては畳込み符号化率 $R=1/2$ であることに対応するためである。

【0036】図7に、ブランチメトリック演算部300に供給されるデータについての、連続する3シンボルとそれらに対応して作成されるブランチメトリックの関係の一例を示す。ここで、最尤状態に対応するブランチメトリックの値を"0"とする。パンクチャドされていない通常のデータが供給される時には、ブランチメトリック演算部300が次のようにしてブランチメトリックBM0, BM1の値を出力する。

【0037】すなわち、例えば極性が'0'で信頼性が高い状態を示す'011'が供給される場合には、状態0である確度を示すメトリックBM0を最尤状態に対応する"0"とし、且つ、状態1である確度を示すメトリックBM1を"7"とする。また、例えば極性が'1'で信頼性が高い状態を示す'100'である場合には、状態0である確度を示すメトリックBM0を"7"とし、且つ、状態1である確度を示すメトリックBM1を"0"とする。1シンボルが新たに入力される毎に、連続する3ビットの組合わせが更新され、それにおうじてBM0, BM1の値が生成される。従って、1シンボルについて、BM0およびBM1の値が1個ずつ生成される。

【0038】このようにして、連続する2シンボルA, Bに対応して、それぞれBM0(A), BM1(A)およびBM0(B), BM1(B)が作成される。これら連続する2シンボルに対応するメトリックに基づいて、ブランチメトリックBM(0,0), BM(0,1), BM(1,0), BM(1,1)が以下のように計算される。

【0039】

BM(0,0), BM(0,1), BM(1,0), BM(1,1)は、それぞれ以下の式(3)',

(4)', (5)' および(6)' になる。

【0041】

$$BM(0,0) = BM0(B) \quad (3)'$$

$$BM(0,1) = BM1(B) \quad (4)'$$

$$BM(1,0) = BM0(B) \quad (5)'$$

$$BM(1,1) = BM1(B) \quad (6)'$$

次に、ACS演算についてより詳細に説明する。上述したように、ステートおよびステート間に生じ得る遷移は、符号化方法に関連して予め規定される。ACS演算は、各ステートに遷移し得る複数のパスの内、最尤のパスを選択する演算処理である。この発明の一実施形態



において使用される畳込み符号を前提とした、3時点間でのステートの遷移について図8を参照して説明する。図8は、ステート00に至る遷移の一例を示すものである。

【0042】各ステートに対して、それぞれ2種類のステートからの遷移が生じ得る。例えば、ステートS00(new)に遷移するパスは、前時点におけるステートS00(old)から来るものと、前時点におけるステートS80(old)から来るものがある。また、ステートS00(old)、ステートS80(old)からステートS00(new)への遷移が生じるためには、入力符号がそれぞれ、(0, 0), (1, 1)である必要があることがわかる。このような状況に鑑みて、一般的には以下のようにして最尤なパスが選択される。

$$S00(new)a = S00(old)a + BM(0, 0) \quad (7)$$

$$S00(new)b = S80(old)b + BM(1, 1) \quad (8)$$

ここで、S00(old)a、S00(old)bは、前時点でのパスメトリックの値である。そして、式(7)、

(8)によって計算されるパスメトリックの値の内か

$$S00(new) = S00(new)a \quad (S00(new)a < S00(new)b \text{ の場合}) \quad (9)$$

$$S00(new) = S00(new)b \quad (\text{それ以外の場合}) \quad (10)$$

式(9)の場合には、前時点でのステートがステート00であるパス(ステート00→ステート00)が最尤であり、一方、式(10)の場合には、前時点でのステートがステート80であるパス(ステート80→ステート00)が最尤であるとされることになる。拘束長 $k=9$ の場合には、状態遷移のパスが全部で $2^{9-1} = 256$ 通りあるので、上述したような計算および選択処理が256ステート分行われる。例として、ステート00に至る遷移(図8に示したもの)およびステート01に至る遷移並びにステート02に至る遷移を表の形式で図9に示す。上述したように、ACS演算部302は、これら2

S00(new)<0> ~ S00(new)<8>の算出結果の中でパスメトリックの値が最小となるものに対応するパスを最尤パスとして選択する。すなわち、以下のようにして選択され

$$S00(new) = \min \{ S00(new)<0>, S00(new)<1>, S00(new)<2>, S00(new)<3>, S00(new)<4>, S00(new)<5>, S00(new)<6>, S00(new)<7> \} \quad (19)$$

そして、このようにして選択された結果を表現する3ビットのパス選択信号が生成される。拘束長 $k=9$ の場合には、全部で256種類のステートに遷移する場合について上述したような最尤パス選択処理を行う。

【0050】なお、ここでは、3時点間に生じるステ

【0043】まず、上述したようにしてブランチメトリックの値を計算する。そして、計算したブランチメトリックの値と、前時点でのパスメトリックの値とに基づいて、新たなパスメトリックの値を算出する。さらに、新たなパスメトリックの値に基づいてパスの尤度を求め、生き残りパスを選択する。また、このような計算を1クロック後のシンボルについても行うために、算出した新たなパスメトリックの値を記憶して保持する。

【0044】新たなステートとしてステート00に遷移する場合を例としてより具体的に説明する。まず、ステート00およびステート80からの遷移に対応する新たなパスメトリックの値S00(new)a、S00(new)bの値が次式のように計算される。

【0045】

$$S00(new)a = S00(old)a + BM(0, 0) \quad (7)$$

$$S00(new)b = S80(old)b + BM(1, 1) \quad (8)$$

ら、以下のようにしてより尤度の高いパスに対応するパスメトリックの値S00(new)が選択される。

【0046】

$$S00(new) = S00(new)a \quad (S00(new)a < S00(new)b \text{ の場合}) \quad (9)$$

$$S00(new) = S00(new)b \quad (\text{それ以外の場合}) \quad (10)$$

56ステート分のパス選択結果を示すパス選択情報を生成する。

【0047】そして、3時点分のデータに基づくACS演算が行われる。かかる演算においては、 $2^3 = 8$ 個のパスメトリック演算結果の内から1個の最尤パスが選択される。具体例として、新ステートとしてステート00に遷移する場合には、以下のような計算によって8種類の旧ステートを起点として、3時点の間に生じ得る8種類のパスについてのパスメトリックの値S00(new)<0> ~ S00(new)<8>が算出される。

【0048】

$$S00(new)<0> = S00(old) + BM[0](0, 0) + BM[1](0, 0) + BM[2](0, 0) \quad (11)$$

$$S00(new)<1> = S20(old) + BM[0](0, 0) + BM[1](1, 0) + BM[2](1, 1) \quad (12)$$

$$S00(new)<2> = S40(old) + BM[0](1, 0) + BM[1](1, 1) + BM[2](0, 0) \quad (13)$$

$$S00(new)<3> = S60(old) + BM[0](1, 0) + BM[1](0, 1) + BM[2](1, 1) \quad (14)$$

$$S00(new)<4> = S80(old) + BM[0](1, 1) + BM[1](0, 0) + BM[2](0, 0) \quad (15)$$

$$S00(new)<5> = S40(old) + BM[0](1, 1) + BM[1](1, 0) + BM[2](1, 1) \quad (16)$$

$$S00(new)<6> = S00(old) + BM[0](0, 1) + BM[1](1, 1) + BM[2](0, 0) \quad (17)$$

$$S00(new)<7> = S00(old) + BM[0](0, 1) + BM[1](0, 1) + BM[2](1, 1) \quad (18)$$

るパスメトリックの値S00(new)に対応するパスが最尤パスとされる。

【0049】

トの遷移を選択するACS演算について説明したが、例えば2時点間に生じるステートの遷移を選択するACS演算を行う場合にも、この発明を適用することができる。2時点間に生じるステートの遷移の例として、ステート00、01、02に至る遷移を図10に示す。な

お、このような場合にもステート数は拘束長によって決まり、例えば拘束長が9の場合には、ステートの総数は256となる。

【0051】パス選択情報記憶部304がパス選択情報を記憶する。また、上述した256ステートの各々に対応するパスメトリックの内から、最尤検出部303が最尤のステートを選択する。すなわち、最も数値が小さいパスメトリックに対応するステートを選択して、そのステートに対応する最尤ステート番号をデータ推定部305に供給する。データ推定部305では、最尤ステート番号を参照してパス選択情報記憶部304の記憶内容を適宜読出してトレースバックを行うことによって最尤復号を行い、復号データを生成する。

【0052】トレースバックについて説明する。まず、処理の開始に際して、最尤ステート検出部303が生成する最尤ステート番号に対応するステートを初期ステートとして設定する。そして、この初期ステートを参照してパス選択情報記憶部304からパス選択情報を読み出す。すなわち初期ステートに到達するパスについてのパス選択情報を読み出し、前ステートのステート番号を得る。その後、さらに、前ステートに至るパスについてのパス選択情報を読み出し、前ステートのさらに前のステート（従って初期ステートから2時点前のステートのステート番号）を得る。以下、同様な処理をパスメモリ長分繰返し、一番古いパス選択情報にまで遡った時点のデータを復号データとする。このような処理をトレースバックと称する。

【0053】この発明は、パンクチャド処理の有無に関連してトレースバックを行うか否かを制御するようにしたものである。すなわち、パンクチャドされていないデータタイミングにおいてトレースバックを行い、復号データを作成する。上述した式(2)に従うパンクチャド処理においては、3回のブランチメトリック作成に対して1回の割合でパンクチャド処理が不必要となる。このため、3回のブランチメトリック作成に対して1回の割合でトレースバックを行い、トレースバックによって一番古いパス選択情報に遡った時点で、その一番古いパス選択情報に基づいて得られる復号データ値と、一番古いパス選択情報よりも1時点および2時点前にトレースバックによって到達するパス選択情報に基づいて得られる復号データ値との、計3ビットのデータを復号データとして出力する。

【0054】また、最後のACS演算が終了した時点以降においては、畳込み符号化の際のテールビットの設定の有無に応じた処理が行われる。すなわち、図4等を参照して上述したように、この発明の一実施形態では畳込み符号化の際に8ビットのテールビットが設定されているので、ステート00を初期ステートとしてトレースバックを開始し、1時点分古い情報にさかのぼる毎に選択したパス選択情報を復号データとする。但し、テールビ

ットが設定されていない場合には、最後のACS演算が終了した時点でその時点での最尤ステート番号を初期ステートとしてトレースバックを開始すれば良い。

【0055】さらにまた、フレーム完結で畳込み符号化を行う場合には、ACS演算におけるパスメトリックの初期値が以下のように設定される。この場合には畳込み符号器の初期値として'0'が設定されるので、畳込み符号が処理の最初に出力するデータが'0'の連続となる。かかる状況に対応して、ステート00に対してはACS演算におけるパスメトリックの初期値として'0'が使用される。また、他のステートに対しては、ACS演算におけるパスメトリックの初期値として、'0'のシンボルデータが連続入力された時の各ステートに対するパスメトリックの値を参照して重み付けがなされた値が使用される。

【0056】次に、図11を参照して、この発明の一実施形態におけるビタビ復号の手順について説明する。ステップS1として、ビタビ復号開始時にパスメトリックの初期値を設定する。次に、ステップS2に移行して、ブランチメトリック演算部300において2シンボルを処理単位としてブランチメトリックを作成し、ACS演算部302において新しいパスメトリック情報およびパス選択情報を作成する。

【0057】この際のブランチメトリック作成は、パンクチャド制御部210から供給される、パンクチャドされたデータタイミングを参照してなされる。そして、新しいパスメトリック情報をパスメトリック記憶部301に記憶すると共に、パス選択情報記憶部304に新しいパス選択情報を記憶する。

【0058】ステップS3においては、ACS演算の回数が（パスメモリ長+テールビット長）よりも小さいか否かが判定される。ACS演算の回数が（パスメモリ長+テールビット長）よりも小さいと判定される場合にはステップS2に移行し、それ以外の場合には、ステップS4に移行する。これにより、ACS演算の回数が（パスメモリ長+テールビット長）分実行された後に、ステップS4に移行するようになされる。ACS演算は、パンクチャドされていないデータタイミングに対応して（この場合にはブランチメトリック作成3回につき1回）行われる。

【0059】ACS演算の回数が（パスメモリ長+テールビット長）を越えてからは、最後のACS演算が終了するまでは、ステップS1以降の処理を行う。まず、ステップS1により、パンクチャドされていないデータタイミングにおいてのみステップS1以降の処理に移行するようになされる。これにより、式(2)に示したようなパンクチャリング・パターンの下では、ブランチメトリック作成3回につき1回の割合でデータ推定部305におけるトレースバックが行われることになる。

【0060】ステップS12においては、最尤検出30

3で検出した最尤ステートを、トレースバック初期アドレスすなわちトレースバックを開始する際の初期ステートとして設定する。さらに、ステップS13として、前の時点のパス選択情報、すなわち前回のACS演算の結果であるパス選択情報をパス選択情報304から読出す。

【0061】そして、ステップS14としてトレースバック回数が(パスメモリ長-2)より小さいか否かを判定する。トレースバック回数が(パスメモリ長-2)より小さいと判定される場合にはステップS17に移行し、それ以外の場合にはステップS15に移行する。ステップS17では、ステップS13において読出したパス選択情報を使用して新トレースバックアドレスを作成し、さらにステップS12に移行することによってトレースバックを繰り返す。

【0062】一方、ステップS15では、復号データを出力する。さらに、ステップS16に移行する。ステップS15では、トレースバック回数がパスメモリ長に等しいか否かが判定される。トレースバック回数がパスメモリ長に等しいと判定される場合にはステップS2に移行し、それ以外の場合はステップS17に移行する。

【0063】以上のような手順により、トレースバック回数がパスメモリ長-2、パスメモリ長-1およびパスメモリ長に等しい場合にそれぞれ復号データが出力されることになる。この結果として、ステップS1以降の処理、すなわちバンクチャドされていないデータタイミングにおいてのみ行われるトレースバックによって、バンクチャドされていないデータタイミングにおけるシンボルに対応する2ビットの復号データをも含めた計3ビットの復号データが出力される。また、最終的にトレース回数がパスメモリ長と等しくなる時にトレースバックが終了し、ブランチメトリック作成、ACS演算、パス選択情報記憶の動作が再度行われることになる。

【0064】また、ステップS4において最後のACS演算が終了したと判定される場合には、ステップS21に移行する。上述したように、この発明の一実施形態では、畳込み符号化において連続する8ビットの'0'からなるテールビットが付加されるので、ステップS21においては、最尤検出部303が生成する最尤ステート情報に基づいてトレースバック初期アドレスを設定するのではなく、テールビットに対応してステート00がトレースバック初期アドレスとして設定される。従って、ステップS21以降の処理においては、データ推定部305においてステート00をトレースバック初期アドレスとするトレースバックが行われる。

【0065】なお、上述したように、テールビットとして例えば連続する'1'等、連続する8ビットの'0'以外のものを使用しても良い。このような場合においても、ステップS21においては、付加されたテールビットに対して最適なステートをトレースバック初期アドレスと

して設定すれば良い。

【0066】すなわち、ステップS22は、パス選択情報記憶部304から前の時点のパス選択情報を読出すステップである。さらに、ステップS23に移行して復号データ1ビットを出力する。そして、ステップS24に移行してトレースバック回数がパスメモリ長と等しいか否かを判定する。トレースバック回数がパスメモリ長と等しい場合には処理を終了し、それ以外の場合には、ステップS25に移行する。ステップS25では、復号データを使用して新トレースバックアドレスを作成し、ステップS22に移行する。このように、ステップS21以降の処理により、トレースバックおよび復号データ出力がトレースバック回数がパスメモリ長に達するまで繰り返される。

【0067】また、フレーム完結で畳込み符号化を行う場合には、新たなフレーム内のデータに対する畳込み符号化を開始するに際して固定値'0'が所定個数(例えば畳込み符号器のレジスタの段数)連続して入力されて畳込み符号器が初期化される。かかる事情に鑑みて、ACS演算におけるパスメトリックの初期値として、ステート00に対しては0、他のステートに対しては'0'のシンボルデータが連続入力された時の各ステートに対する値をそれぞれ使用するようにしても良い。すなわち、畳込み符号器の初期化に則して重み付けされた値をACS演算におけるパスメトリックの初期値として設定するようにしても良い。

【0068】一方、上述したように、ステップS2におけるACS演算の可否を、トレースバックの可否と同様に、バンクチャド処理の有無に関連して制御するようにしても良い。すなわち、バンクチャド処理がなかった時点でACS演算とトレースバックを行い、復号データを作成する。従って、式(2)に示したようなバンクチャリング・パターンの下では、ブランチメトリック作成3回につき1回の割合でACS演算とトレースバックとを行い、トレースバック時に一番古い情報にさかのぼった時点とそれよりも2回前の時点の情報までの計3ビットのデータを復号データとする。

【0069】上述したこの発明の一実施形態においては、トレースバックを行う際の初期ステート(トレースバック初期アドレス)がACS演算の結果に基づいて検出される最尤ステートとされるか、または、テールビットの付加を考慮して所定の条件の下でステート0とされる。これに対して、システムの特性等を考慮して、初期ステートとして他のステートを設定するようにしても良い。

【0070】また、この発明の一実施形態では、畳込み符号の拘束長 $K=9$ 、符号化率 $R=1/2$ としたが、拘束長 $K$ 、符号化率 $R$ が他の値をとるような畳込み符号を用いる場合にも、この発明を適用することができる。

【0071】また、この発明の一実施形態は、移動体通

信機にこの発明を適用したものであるが、この発明は、パンクチャドビタビ復号を行う情報伝送装置に適用することが可能である。

#### 【0072】

【発明の効果】上述したように、この発明は、パンクチャドビタビ復号方法において、パンクチャドされたデータタイミングを検出し、かかるデータタイミングに対応するシンボルに基づいて初期値を設定してトレースバックを行うことをせず、パンクチャドされていないデータタイミングに対応するシンボルに基づいて初期値を設定してトレースバックを行って復号データを生成するようにし、その際にパンクチャドされているデータタイミングに対応するシンボルに基づく復号データをも出力するようにしたものである。

【0073】このため、パンクチャドされたデータに基づく初期設定の下でのトレースバックが行われなくすることができる。従って、そのようなトレースバックが行われる結果として畳込み符号中の誤りに起因して最尤パスとして誤ったものが選られ、それによって復号誤りが生じることを防止または低減することができる。

【0074】これにより、最尤ステートが正しく選択される確率を高くするためにトレースバックを行うパスメモリの長さを長く設定すること無く、ビタビ復号器による復号特性を向上させることが可能となる。

【0075】従って、パスメモリの長さを長くすることに起因する問題点の発生を伴うことなく復号エラーレートの低減を実現することができる。すなわち、クロックを高速化しなくても、復号処理に要する時間を短縮することができ、デコード遅延時間を小さくすることができる。また、復号処理に係る構成の回路規模を小さくすることができる。

【0076】また、フレーム等の所定量のデータ毎に処理が完結し、処理が完結する毎に畳込み符号器を初期化するようにした畳込み符号化によって生成されるデータを復号する際に、ACS演算を開始する際の各初期ステートのパスメトリック値として畳込み符号器の出力が例えば'0'等の固定値の連続となる場合に計算される各ステートのパスメトリック値を参照して重み付けがなされた値を設定するようにすれば、各ステートについてのパスメトリック値をより確度良く算出できる。このため、選択されるパスの尤度を向上させることができる。

【0077】また、所定量のデータ毎に処理が完結し、

処理が完結する毎に符号化したデータの最後にテールビットを付加するようにした畳込み符号化によって生成されるデータを復号する際に、復号すべき残りのデータの量がトレースバック長よりも小さくなった時点以降において、テールビットに対応するステートを起点としてトレースバックを行うようにすれば、復号すべきデータの内の終端付近において精度の高いトレースバックを行うことが可能となる。

#### 【図面の簡単な説明】

10 【図1】この発明の一実施形態の全体的な構成について説明するためのブロック図である。

【図2】この発明の一実施形態の一部の構成についてより詳細に説明するためのブロック図である。

【図3】この発明を適用することができる送受信系において使用される畳込み符号データフォーマットの一例について説明するための略線図である。

【図4】この発明の一実施形態において、データの送信／受信に係る処理についてより具体的に説明するためのブロック図である。

20 【図5】畳込み符号器の構成の一例を示すブロック図である。

【図6】この発明の一実施形態中のビタビ復号器の構成について説明するためのブロック図である。

【図7】この発明の一実施形態においてブランチメトリックを生成する方法について説明するための略線図である。

【図8】3時点間でのステートの遷移について説明するための略線図である。

30 【図9】3時点間でのステートの遷移の例を示す略線図である。

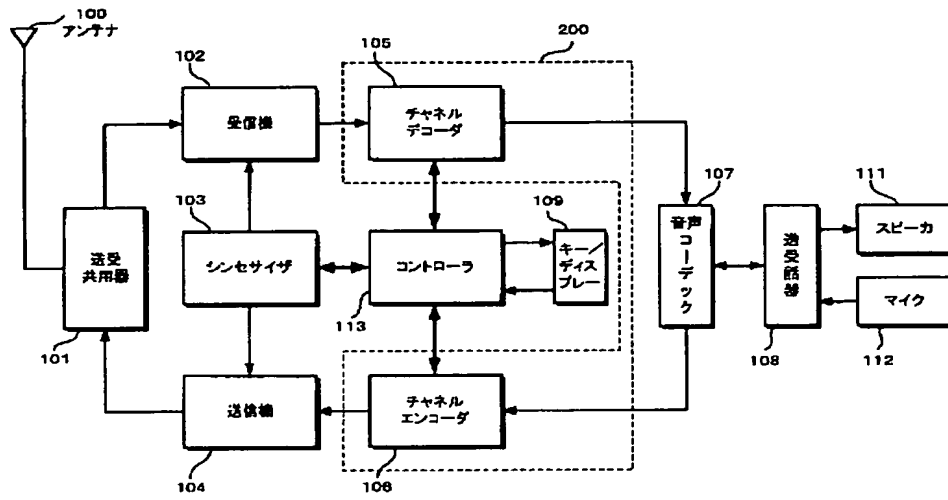
【図10】2時点間でのステートの遷移の例を示す略線図である。

【図11】この発明の一実施形態におけるパンクチャドビタビ復号方法の手順について説明するための略線図である。

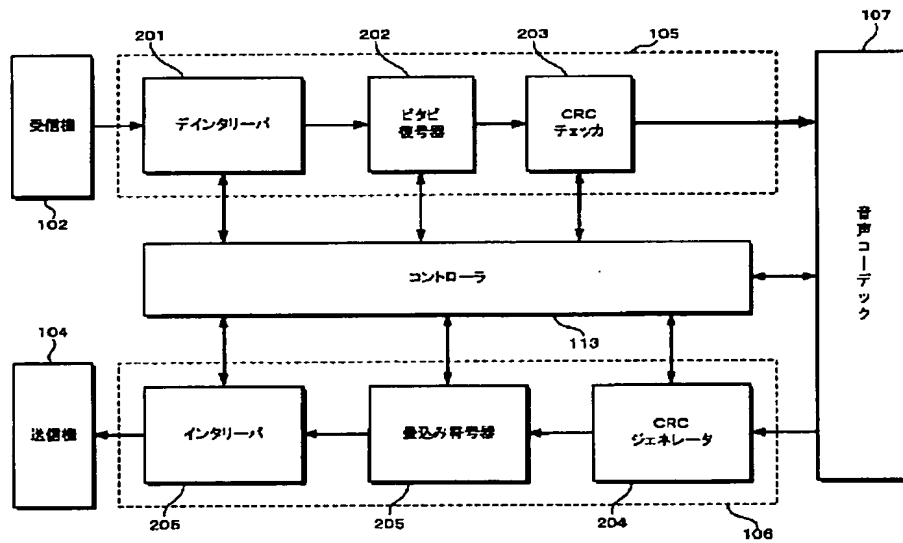
#### 【符号の説明】

105・・・チャネルデコーダ、106・・・チャネルエンコーダ、202・・・ビタビ復号器、205・・・畳込み符号器、207・・・パンクチャド処理部、210・・・パンクチャド制御部、300・・・ブランチメトリック演算部、302・・・ACS演算部、305・・・データ推定部

【図 1】



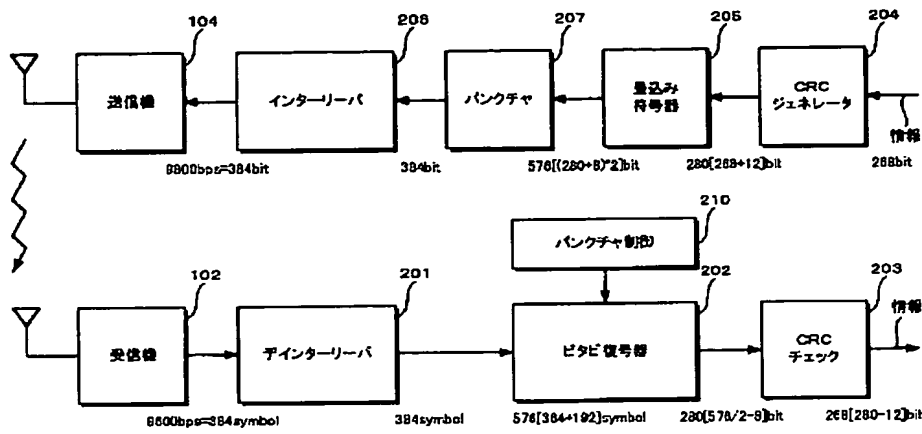
【図 2】



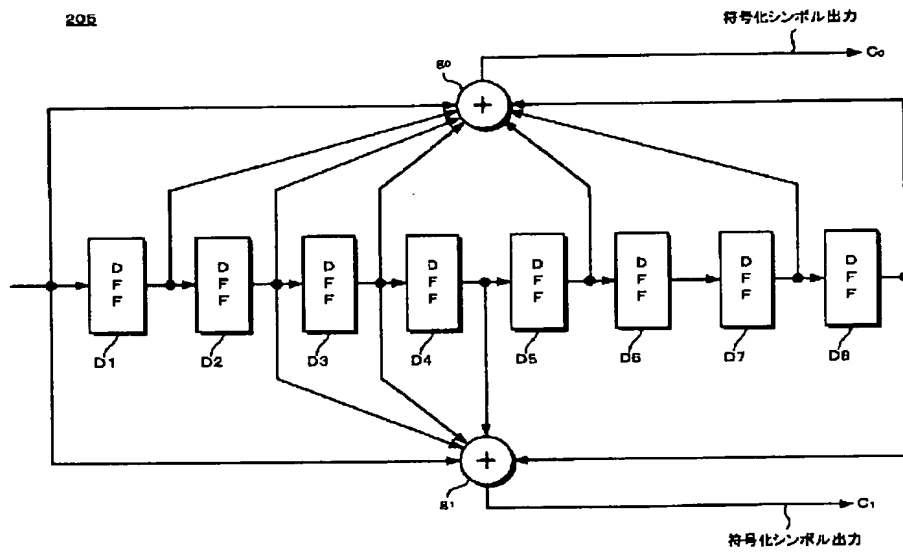
【図 3】

伝送速度	インタープ 後シンボル数	バンクチャ	畳込み符号化 後シンボル数	畳込み 符号化率	データ量 /20mssec	情報	CRC	Encoder Tail
9600 bps	384 symbol	4/8	576 symbol	1/2	288 bit	288 bit	12 bit	8bit

【図 4】



【図 5】



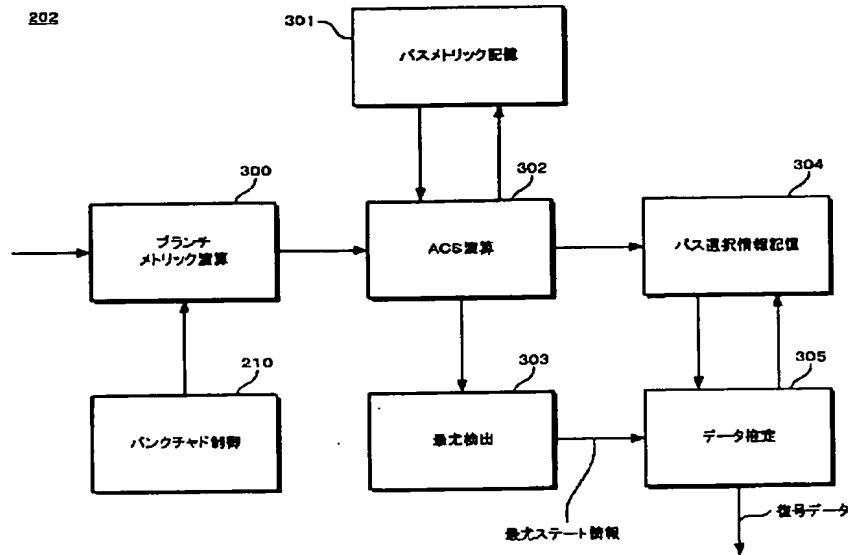
【図 7】

Symbol data			補正	信頼性	Metric(Max)	
bit2	bit1	bit0			BM0	BM1
0	1	1	0	High	0	7
0	1	0	0		1	8
0	0	1	0		2	5
0	0	0	0	Low	3	4
1	1	1	1	Low	4	3
1	1	0	1		6	2
1	0	1	1		6	1
1	0	0	1	High	7	0

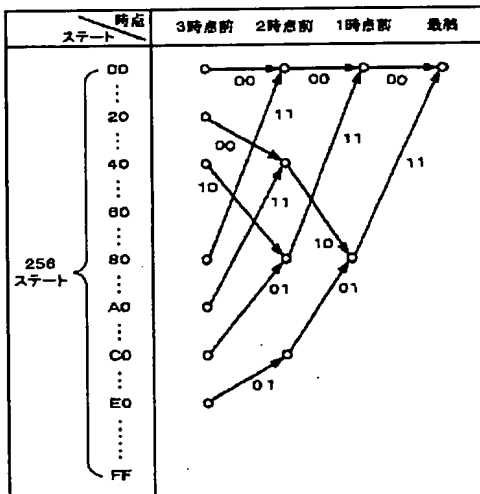
【図 10】

2時点前の ステート	1時点前の ステート	最終ステート
00	00	00
40	80	00
80	00	00
C0	80	00
00	00	01
40	80	01
80	00	01
C0	80	01
00	01	02
40	81	02
80	01	02
C0	81	02

【図 6】



【図 8】



【図 9】

3時点前の ステート	2時点前の ステート	1時点前の ステート	最終ステート
00	00	00	00
20	40	80	00
40	60	00	00
60	C0	80	00
80	00	00	00
A0	40	80	00
C0	80	00	00
E0	C0	80	00
00	00	00	01
20	40	80	01
40	80	00	01
60	C0	80	01
80	00	00	01
A0	40	80	01
C0	80	00	01
E0	C0	80	01
00	00	01	02
20	40	81	02
40	80	01	02
60	C0	81	02
80	00	01	02
A0	40	81	02
C0	80	01	02
E0	C0	81	02

【図11】

